Revisiting Cross-View Completion: Self-Supervised Pre-Training via Reconstruction Error Comparison

Thibaut Loiseau[1] **Vincent Lepetit**[1] **Guillaume Bourmaud**[2]
[1] LIGM, Ecole Nationale des Ponts et Chaussées, IP Paris, Univ Gustave Eiffel, CNRS, France
[2] Univ. Bordeaux, CNRS, Bordeaux INP, IMS, UMR 5218, France
{thibaut.loiseau,vincent.lepetit}@enpc.fr guillaume.bourmaud@u-bordeaux.fr

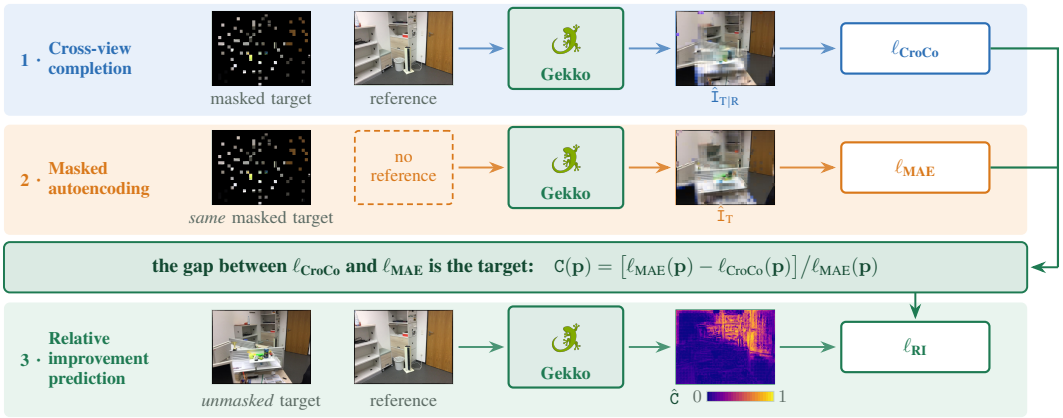


Figure 1: **Gekko: one network, three passes, no 3D annotation.** A single network F is run three times on the same video frame pair: **(1)** it reconstructs a 90%-masked target *with* the reference view, **(2)** it reconstructs the *same* masked target *without* it, and **(3)** it regresses their per-pixel gap $C(\mathbf{p})$ from the *unmasked* pair through one extra output channel $\hat{C}$. That gap is large where a pixel is co-visible and vanishes where it is not, with no depth, pose or label anywhere in the pipeline. Regressing it is what turns cross-view completion's blind spot into supervision: it supplies a binocular training signal for *all* masked regions, including the non-co-visible ones, where CroCo's signal is implicitly monocular.

Project page: thibautloiseau.github.io/projects/gekko

Abstract

Self-supervised pre-training via cross-view completion learns strong features for 3D vision from co-visible regions of image pairs. However, the reference view provides little information for reconstructing non-co-visible patches, implicitly yielding a monocular training signal in these regions. We introduce **Gekko**, which turns this limitation into a useful signal. The *relative improvement* of the cross-view reconstruction error over a masked-autoencoder error is a self-supervised proxy for co-visibility: large improvements indicate co-visible regions, negligible ones non-co-visible areas. Gekko is a network, trained from scratch, that jointly performs cross-view completion, masked autoencoding, and per-pixel prediction of this relative improvement, providing an additional binocular signal for *all* masked regions without any ground-truth 3D annotation. Under identical architectures and training data, Gekko's frozen features consistently outperform CroCo's on zero-shot correspondence estimation, relative pose estimation, and pointmap regression, with up to 6× higher accuracy at the strictest relative-pose threshold and a 22% drop in end-point error on ETH3D, and they also outperform released cross-view backbones of comparable or larger size. The extra channel it learns is itself a strong co-visibility detector on unseen scenes. It can also be trained directly from raw videos with a simple stride-based curriculum, removing the cumbersome 3D preprocessing prior methods require while matching models trained on curated data. Code and pre-trained models are publicly available.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Learning meaningful representations for 3D vision from visual data is a fundamental challenge in computer vision [12, 25, 10, 4, 26, 71, 38, 64], with applications spanning depth estimation [63, 33], visual localization [16], and 3D reconstruction [57, 22, 62, 54]. Self-supervised pre-training has emerged as a powerful paradigm for acquiring these representations without expensive manual annotations. Among recent approaches, cross-view completion [60] extends Masked Autoencoders (MAE) [26] to image pairs: patches in a target image are masked and reconstructed using both the remaining target patches and a reference view. The underlying principle is that reconstructing co-visible regions forces the network to establish correspondences and reason about geometry, yielding features that transfer effectively to downstream 3D tasks [2, 57, 30].

However, cross-view completion faces a critical limitation. When a masked patch is not co-visible (*e.g.*, due to occlusion), the reference view provides little information in those regions, implicitly yielding a MAE-like monocular training signal. Consequently, cross-view completion pre-training does not benefit from image pairs with low overlap, where many masked patches lack a corresponding region in the reference view [60, 36].

We propose **Gekko**, a fully self-supervised pre-training framework that turns this limitation into a strength. Our key insight is that the *relative improvement* of the cross-view completion reconstruction error over the MAE reconstruction error is a reliable proxy for co-visibility (Figure 1). In co-visible regions, incorporating the reference image substantially improves reconstruction quality, yielding a large relative improvement. In non-co-visible regions, both reconstructions perform similarly, and the relative improvement is near zero. This proxy requires no ground-truth depth maps or camera poses, making it applicable to any collection of image pairs, including uncalibrated data.

Gekko is a network that jointly performs three tasks during pre-training: (i) cross-view completion, (ii) masked autoencoding, and (iii) prediction of the relative improvement map. The architecture is identical to CroCo [60] up to a single additional output channel, enabling a direct and fair comparison. The relative improvement prediction provides an additional dense binocular self-supervised signal for *all* masked regions. The architectural cost of this third task is negligible: the encoder and the cross-attention decoder are byte-for-byte those of CroCo, and only the lightweight pixel head grows from three to four channels, so fine-tuning and inference cost exactly what CroCo costs, up to one extra output channel. Pre-training is more expensive, since the extra passes raise activation memory (Section 5.10).

Under identical architectures and training data, Gekko consistently outperforms CroCo across three downstream tasks: up to $6\times$ higher accuracy at the strictest threshold on relative metric pose estimation (ScanNet-1500), a 22% drop in end-point error on zero-shot correspondence estimation (ETH3D), and lower error on pointmap regression, with gains that scale to larger architectures. We also enable fully self-supervised pre-training from raw video with a simple stride-based curriculum, removing the expensive 3D preprocessing prior methods require [60, 61, 36] while matching models trained on curated data. The extra channel is also directly interpretable: probed on its own, it detects co-visibility on scenes never seen during pre-training, and does so better than the cross-view reconstruction error it is derived from (Section 5.6).

Our contributions are as follows:

- We show that the relative improvement of a cross-view completion error over a masked autoencoding error is a reliable self-supervised proxy for co-visibility, substantially outperforming the cross-view completion error alone, and we propose Gekko, a network trained from scratch that jointly learns cross-view completion, masked autoencoding and relative improvement prediction. The relative improvement loss supplies a binocular signal for *all* masked regions without any ground-truth 3D labels.
- We demonstrate that Gekko consistently outperforms CroCo on three cross-view 3D vision tasks under identical architectures and training data, that its predicted channel is itself a strong co-visibility detector on scenes unseen during pre-training, and that its frozen features outperform released cross-view backbones of comparable or larger size.
- We show that Gekko can be trained from raw videos with a simple stride-based curriculum, removing the need for overlap-based data preprocessing and matching the performance of models trained on curated data, enabling fully self-supervised pre-training.

2 Related work

Self-supervised visual representation learning. Masked image modeling [26, 4, 71] and self-distillation [10, 38, 3] have become dominant paradigms for learning visual features without labels. MAE [26] reconstructs randomly masked patches and learns rich visual features, while DINO [10] learns representations through self-distillation, and I-JEPA [3] by predicting masked regions in latent space. DINOv2 [38] combines both strategies at scale. These methods operate on single images and therefore do not explicitly capture multi-view geometric relationships. Our work builds on masked image modeling but extends it to image pairs, where the comparison between monocular and binocular reconstruction errors provides a geometric training signal absent from single-image methods.

Pre-training on image pairs. CroCo [60] pioneered cross-view completion for self-supervised pre-training of 3D vision tasks by extending masked image modeling to image pairs. CroCo v2 [61] scaled this framework to large and diverse training data, demonstrating strong transfer to stereo matching and optical flow. PMatch [73] introduces a variant where both images are partially masked to pre-train an image matching model. Other variants target object tracking, through masked appearance transfer [68] or correlative masked modeling [49]. Notably, the foundational models DUSt3R [57] and MASt3R [30], and all subsequent works [62, 9, 74, 16, 67, 48, 22], build upon CroCo's cross-view completion framework and have achieved remarkable success in dense 3D reconstruction and matching. All these methods share a common limitation: the training signal is implicitly monocular for non-co-visible regions. Our method addresses this by explicitly modeling the relative improvement of cross-view completion over monocular reconstruction, providing an additional binocular signal for *all* masked regions. A parallel line of work extends cross-view completion beyond image pairs: MuM [37] masks arbitrarily many views uniformly and decodes them with inter-frame attention, and Muskie [31] is a native multi-view backbone that learns view-invariant features by finding correspondences across views without 3D supervision. These extensions are orthogonal to our contribution (Gekko improves the training signal available within each pair), and combining the two is a promising direction, though not a straightforward one: with more than two views, the relative improvement might have to be defined against a *set* of references instead of a single one (Section 5.10).

Co-visibility in image matching and pre-training. Ground-truth co-visibility masks, derived from depth maps and camera poses, are widely used in image matching methods [39, 19, 34, 69, 50, 59, 20, 11, 56, 52, 18, 23, 29, 24, 27, 51, 43] to compute overlaps and select training pairs. Several approaches [43, 20, 21, 53, 18, 24, 6] also leverage these masks to learn matchability scores at test time. Alligat0R [36] recently demonstrates that co-visibility prediction is an effective *supervised* pre-training objective for relative pose, training on ground-truth co-visibility labels derived from depth maps and camera poses. While effective, this approach requires ground-truth 3D annotations and expensive overlap-based data preprocessing. Our method pursues the same goal in a self-supervised manner: the relative improvement of cross-view completion over masked autoencoding serves as a proxy for co-visibility, eliminating the need for any ground-truth depth or pose information. Combined with a simple stride-based curriculum (Section 5.4), Gekko can be trained directly from raw videos, removing the cumbersome 3D data preprocessing required by both CroCo and Alligat0R, thereby enabling fully self-supervised pre-training.

3 Background and motivation

3.1 Background on CroCo and MAE

CroCo [60] reconstructs a target image $\mathtt{I}_\mathrm{T}$ from a masked version of itself and a reference image $\mathtt{I}_\mathrm{R}$, using a ViT encoder, a cross-attention decoder, and a pixel decoder (full formalism in Section B). The per-pixel reconstruction error on masked pixels is:

$$\ell_{\text{CroCo}}(\mathbf{p}) = \left\| \mathtt{I}_\mathrm{T}(\mathbf{p}) - \hat{\mathtt{I}}_{\mathrm{T|R}}(\mathbf{p}) \right\|_2^2 . \tag{1}$$

In co-visible regions CroCo transfers information from the reference, implicitly learning correspondences [2]. Elsewhere the reference adds little and the reconstruction degrades (Figure 4).

A Masked Autoencoder (MAE) [26] reconstructs $\mathtt{I}_\mathrm{T}$ from a masked version of itself *only*, with per-pixel error:

$$\ell_{\mathrm{MAE}}(\mathbf{p}) = \left\| \mathtt{I}_\mathrm{T}(\mathbf{p}) - \hat{\mathtt{I}}_\mathrm{T}(\mathbf{p}) \right\|_2^2. \tag{2}$$

MAE cannot exploit the reference view at all: it inpaints from unmasked target patches alone.

3.2 Relative improvement of CroCo over MAE

A natural question is whether co-visibility can be predicted from CroCo's reconstruction error alone. Figure 2 (left) shows histograms of $\ell_{\mathrm{CroCo}}(\mathbf{p})$ split by ground-truth co-visibility labels. The distributions largely overlap, indicating that CroCo's error alone is an unreliable predictor.

MAE provides a reconstruction error baseline that does not exploit cross-view cues and is independent of the co-visibility between the reference and target images.

This raises another question: *can co-visibility be predicted by comparing CroCo's and MAE's reconstruction errors?*

We define the relative improvement of CroCo over MAE:

$$\mathtt{C}(\mathbf{p}) = \frac{\ell_{\mathrm{MAE}}(\mathbf{p}) - \ell_{\mathrm{CroCo}}(\mathbf{p})}{\ell_{\mathrm{MAE}}(\mathbf{p})}. \tag{3}$$

In co-visible regions, $\ell_{\mathrm{CroCo}}(\mathbf{p}) \ll \ell_{\mathrm{MAE}}(\mathbf{p})$ and consequently $\mathtt{C}(\mathbf{p}) \approx 1$, because the network has access to the reference view. In non-co-visible regions, both networks perform similarly, leading to $\mathtt{C}(\mathbf{p}) \approx 0$.

Figure 2 (right) reports histograms of $\mathtt{C}(\mathbf{p})$ computed using ground-truth co-visibility labels. Ideally, co-visible regions would concentrate near 1. The distribution does spread over $[0, 1]$, and the reason is structural: $\mathtt{C}(\mathbf{p})$ is a ratio whose denominator is the MAE error, so it is uninformative wherever that error is already small. This happens in uniform regions but equally in repeated or self-similar structure: the target alone already predicts the pixel well, the reference cannot improve on it, and $\mathtt{C}(\mathbf{p})$ stays near zero whether or not the pixel is co-visible. $\mathtt{C}(\mathbf{p})$ therefore *approximates* co-visibility rather than classifying it, and $\ell_{\mathrm{MAE}}(\mathbf{p})$ is itself the detector of when it fails, which is what Section 4.3 exploits. Nevertheless, compared to CroCo's error alone (Figure 2 (left)), the two distributions overlap substantially less, reaching an average precision of 0.74 compared to 0.57 for co-visibility prediction. The relative improvement $\mathtt{C}(\mathbf{p})$ is thus a more reliable indicator of co-visibility. In Section 5.6 we show that the channel Gekko actually *learns* is a substantially better co-visibility classifier still, on scenes never seen during pre-training.

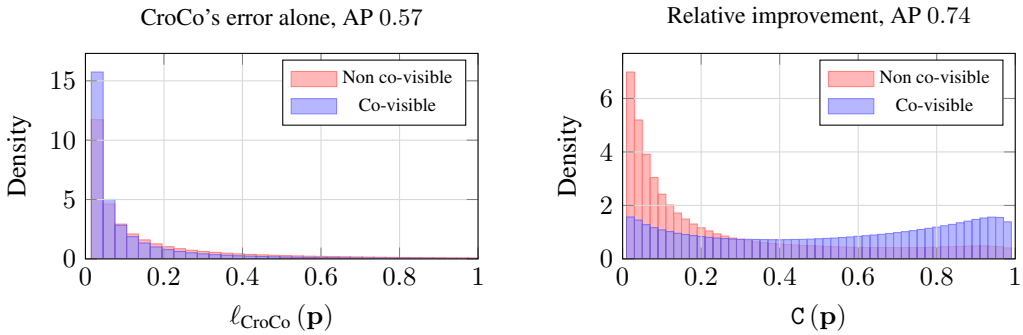


Figure 2: **The relative improvement of CroCo over MAE separates co-visibility.** Histograms split by ground-truth co-visibility, computed on the same 100 ScanNet-50 pairs (Section O). **(left)** CroCo's reconstruction error $\ell_{\mathrm{CroCo}}(\mathbf{p})$ (Eq. 1) alone. Errors are on average lower in co-visible regions (blue) than in non-co-visible ones (red), but the two distributions largely overlap, giving an average precision of only 0.57. **(right)** The relative improvement $\mathtt{C}(\mathbf{p})$ (Eq. 3). Values concentrate near zero for non-co-visible regions and spread across $[0, 1]$ for co-visible ones, raising the average precision to 0.74.

4 Method

The relative improvement $\mathtt{C}(\mathbf{p})$ from Section 3.2 provides a pixelwise, albeit noisy, self-supervised co-visibility signal. Predicting co-visibility requires geometric reasoning, making it a useful training objective for 3D vision [36]. However, existing co-visibility objectives rely on ground-truth depth maps and camera poses, limiting their applicability.

We introduce Gekko, a self-supervised pre-training method that uses $\mathtt{C}(\mathbf{p})$ as pseudo co-visibility labels to improve CroCo's pre-training without any 3D annotations (Figure 1).

4.1 Pre-training strategy

Gekko is a network F_{Gekko}, *trained from scratch*, and used in three complementary ways during pre-training.

Given a target image $\mathtt{I}_\mathtt{T}$ and a reference image $\mathtt{I}_\mathtt{R}$ of the same scene from different viewpoints, Gekko performs three forward passes:

1. A cross-view completion pass that reconstructs $\mathtt{I}_\mathtt{T}$ from a masked version of itself *and* $\mathtt{I}_\mathtt{R}$:
$$\hat{\mathtt{I}}_{\mathtt{T}|\mathtt{R}} = F_{\text{Gekko}}\left(\mathtt{M} \odot \mathtt{I}_\mathtt{T}, \mathtt{I}_\mathtt{R}\right), \tag{4}$$
where $\mathtt{M}$ masks out a random subset (90%) of the target input patches.
2. A masked autoencoder pass that reconstructs $\mathtt{I}_\mathtt{T}$ from a masked version of itself *only*:
$$\hat{\mathtt{I}}_\mathtt{T} = F_{\text{Gekko}}\left(\mathtt{M} \odot \mathtt{I}_\mathtt{T}\right), \tag{5}$$
where $\mathtt{M}$ is the *same* mask as the one used for the cross-view completion pass.
3. A relative improvement prediction pass that predicts the relative improvement map $\mathtt{C}$ (Eq. 3) of CroCo's reconstruction error (first pass) over MAE's reconstruction error (second pass), from $\mathtt{I}_\mathtt{T}$ (unmasked) and $\mathtt{I}_\mathtt{R}$:
$$\hat{\mathtt{C}} = F_{\text{Gekko}}\left(\mathtt{I}_\mathtt{T}, \mathtt{I}_\mathtt{R}\right). \tag{6}$$

4.2 Network architecture

The architecture F_{Gekko} follows CroCo [60]: a ViT-based encoder, a cross-attention based decoder, and a lightweight pixel decoder. In its original version, the pixel decoder outputs three channels corresponding to the RGB values of the reconstructed target image. In Gekko, a fourth channel is added to predict the relative improvement map $\hat{\mathtt{C}}$. This architecture handles all three passes: in the MAE pass (second pass), since there is no reference image, the cross-attention layers operate as self-attention with the unmasked target patches and dedicated mask tokens, as in classical MAE. The fact that CroCo and Gekko share the same architecture (up to the extra output channel) and the same training data enables fair comparison in Section 5.

4.3 Training objective

The network F_{Gekko} is trained *from scratch*, in a fully self-supervised manner, minimizing:

$$\mathcal{L}_{\text{Gekko}} = \sum_{\mathbf{p} \in \Omega_\mathtt{M}} \ell_{\text{MAE}}\left(\mathbf{p}\right) + \ell_{\text{CroCo}}\left(\mathbf{p}\right) + \ell_{\text{RI}}\left(\mathbf{p}\right) \tag{7}$$

where $\Omega_\mathtt{M}$ is the set of masked target pixel locations, ℓ_{MAE} and ℓ_{CroCo} are defined in Eq. 2 and Eq. 1 respectively, and the relative improvement loss is:

$$\ell_{\text{RI}}\left(\mathbf{p}\right) = \left(sg\left[\ell_{\text{MAE}}\left(\mathbf{p}\right) - \ell_{\text{CroCo}}\left(\mathbf{p}\right)\right] - sg\left[\ell_{\text{MAE}}\left(\mathbf{p}\right)\right]\hat{\mathtt{C}}\left(\mathbf{p}\right)\right)^2, \tag{8}$$

where $sg\left[\cdot\right]$ denotes the stop-gradient operator. By construction, when $\hat{\mathtt{C}}\left(\mathbf{p}\right) = \frac{\ell_{\text{MAE}}(\mathbf{p}) - \ell_{\text{CroCo}}(\mathbf{p})}{\ell_{\text{MAE}}(\mathbf{p})}$, the loss satisfies $\ell_{\text{RI}}\left(\mathbf{p}\right) = 0$. Note that Eq. 8 differs slightly from a direct regression of $\mathtt{C}(\mathbf{p})$. This formulation down-weights pixels where the MAE loss is low (see Section G), which often correspond to uniform regions that yield noisy pseudo-labels. This down-weighting leads to significantly improved performance and sharper co-visibility maps (see Section 5.8 and Figure 3). As in CroCo [60], reference and target images are normalized such that each patch has zero mean and unit variance.

In Figure 3, Gekko's relative improvement maps $\hat{\mathtt{C}}$, while less sharp in uniform regions, are strongly correlated with ground-truth co-visibility labels.

5 Experiments

We evaluate Gekko on three complementary tasks: zero-shot correspondence estimation directly after pre-training following [2], relative metric pose estimation after fine-tuning with a lightweight prediction head on top of frozen features, and pointmap regression with a DPT head [40]. All comparisons against CroCo use identical architectures and training data to ensure fairness. We also test generality across architectures (Section 5.9). The appendix reports single-image probes on depth, semantic segmentation, classification and Taskonomy(Section L), CroCo's optical flow and 7-Scenes pose benchmarks with frozen and fully fine-tuned backbones (Section M), complete pointmap metrics (Section N), data scaling on the raw-video mix (Section K), a per-row ablation analysis (Section F), and additional qualitative results (Section Q).

5.1 Experimental setup

Pre-training datasets. All models are pre-trained on the indoor scenes ScanNet split of the Cub3 dataset [36], which consists of image pairs extracted from ScanNet [13]. Two overlap variants test whether Gekko benefits from low-overlap pairs, which standard cross-view completion does not: at least 50% overlap (*ScanNet-50*) and at least 5% (*ScanNet-all*). In Section 5.4, models are also pre-trained on DL3DV [35] to evaluate generalization to outdoor and diverse scenes. Section 5.6 and Section 5.7 instead use a 12-source mix of raw video, consumed with the stride curriculum and no 3D preprocessing (Section H). Those models are written Gekko-B$^{\text{mix}}$ and Gekko-L$^{\text{mix}}$.

Model architectures. Following CroCo [60], all models use a ViT-based architecture [17] in Base and Large sizes (details in Section D). As described in Section 4.2, Gekko adds a single output channel to CroCo's architecture to predict the relative improvement map $\hat{\mathbf{C}}$, enabling direct and fair comparison. All pre-training and fine-tuning hyperparameters are given in Section C. Every CroCo result reported in this paper is our own pre-training from scratch with the same architecture, data, augmentation and schedule. No released checkpoint is used, with the single exception of the publicly released backbones probed in Table 7.

5.2 Zero-shot correspondence estimation

Features are first evaluated immediately after pre-training, without any fine-tuning, on dense correspondence estimation. Following [2], we evaluate on ETH3D [44] and report the Average End-Point Error (AEPE, ↓) for correspondences read from either the encoder or the decoder, both of which cross-view completion models implicitly learn. Gekko cuts AEPE by 22% for encoder features and 21% for decoder features over the CroCo baseline (Table 1), so the extra training signal improves features for downstream tasks relying on either part of the network.

Table 1: **Zero-shot correspondences on ETH3D.** AEPE (↓). All models are Base, pre-trained on ScanNet-50.

Model	AEPE (↓)
Encoder	
CroCo-B	78.4
Gekko-B (ours)	**60.8**
Decoder	
CroCo-B	101.9
Gekko-B (ours)	**80.8**

Table 2: **Curriculum pre-training on ScanNet-1500.** Percentage of pairs (↑) within thresholds. "Curriculum" denotes training from raw videos with a stride-based schedule (Section 5.4). Best in **bold**.

Pre-training data	Model	ScanNet-1500		
		10°/0.25m	10°/0.5m	10°/1m
ScanNet (curriculum)				
50	CroCo-B	20.1	34.4	40.2
	Gekko-B	31.2	45.0	49.0
all	CroCo-B	18.9	35.1	43.9
	Gekko-B	**39.4**	**56.5**	**61.1**
DL3DV (curriculum)				
50	CroCo-B	8.5	19.9	27.2
	Gekko-B	29.0	48.4	54.3
all	CroCo-B	15.6	29.3	36.8
	Gekko-B	**34.9**	**51.1**	**57.1**

5.3 Relative metric pose estimation

We next fine-tune a lightweight MLP head on frozen features for relative *metric* pose estimation on ScanNet-1500 [43], a task that requires reasoning about both viewpoint and scene geometry. Gekko-B outperforms CroCo-B by more than $5\times$ at the strictest threshold (Table 3), and the gap widens on ScanNet-all (43.7% vs. 6.6%): Gekko benefits from low-overlap pairs where CroCo does not.

The advantage carries over to Large models: on ScanNet-all, Gekko-L reaches 51.5% at $10°$/0.25m against 7.5% for CroCo-L, a $6.9\times$ improvement. Both Large models are pre-trained by us for 100k steps at the same global batch size, so the comparison isolates the training objective at both scales. The benefits of the relative improvement loss thus carry over to larger architectures.

Table 3: **Relative metric pose estimation on ScanNet-1500.** Percentage of pairs (↑) within rotation/translation thresholds. Every row is our own pre-training from scratch. Within each of the Base and Large groups the two rows share architecture, pre-training data, global batch size (768) and number of steps (100k), so the only difference is the training objective. Best result per column in **bold**.

Model	ScanNet-50			ScanNet-all		
	10°/0.25m	10°/0.5m	10°/1m	10°/0.25m	10°/0.5m	10°/1m
CroCo-B	5.5	12.9	18.5	6.6	15.7	23.0
Gekko-B (ours)	**29.1**	**43.3**	**47.5**	43.7	54.4	57.6
CroCo-L	–	–	–	7.5	16.1	23.0
Gekko-L (ours)	–	–	–	**51.5**	**64.3**	**68.4**

5.4 Curriculum pre-training

3D preprocessing [61, 36] is expensive and limits scalability to new datasets. We therefore train directly from raw video with a simple stride-based curriculum.

Curriculum strategy. Pairs start at a stride of 1 between consecutive frames: every N steps the maximum stride $k_{\max}$ grows by Δk, and each pair draws its stride uniformly from $[1, k_{\max}]$. Two variants mimic the overlap distributions of the preprocessed datasets: a *-50* variant producing mostly high-overlap pairs, and an *-all* variant that reaches low-overlap pairs earlier. Schedule constants are given in Section C.

Results. Table 2 reports pose estimation on ScanNet-1500. Trained from raw video with no 3D annotation and no overlap-based preprocessing, Gekko matches its curated-data counterpart (31.2% vs. 29.1% at $10°$/0.25m. The exact training pairs differ between the two settings). The curriculum helps CroCo considerably on its own (20.1% vs. 5.5%), so progressive stride scheduling is useful independently of our objective. Yet Gekko keeps a clear margin under it, on DL3DV too. Better sampling does not substitute for the relative improvement signal.

5.5 Pointmap regression

Following DUSt3R [57], a DPT head on frozen features regresses per-pixel 3D pointmaps in the first image's frame, evaluated in-domain (ScanNet) and out-of-domain (DL3DV, ETH3D). Gekko outperforms CroCo on all three benchmarks and both pre-training variants (Table 4), reducing Chamfer error by 10% in-domain and by 5–7% out-of-domain, so the relative improvement signal produces features that generalize beyond the fine-tuning distribution.

5.6 Does the predicted channel recover co-visibility?

Section 3.2 showed that the *analytic* relative improvement separates co-visible pixels better than ℓ_{CroCo} alone. We now test what the network actually predicts (Tables 5 and 6): we run Gekko once per pair with *no masking* and threshold $\hat{\mathtt{C}}$ as a per-pixel co-visibility classifier, against the same ground truth as Section 3.2. Both models are Base, 200k steps on the mix of Section H. The protocol is in Section I.

Table 4: **Pointmap regression on ScanNet, DL3DV and ETH3D.** Chamfer Overall (↓) for models pre-trained on DL3DV with curriculum and fine-tuned on ScanNet-all. Gekko outperforms CroCo on all three benchmarks, including out-of-domain evaluation on DL3DV and ETH3D. Accuracy and Completeness are broken out in Table 14. Best per group in **bold**.

Pre-training	Model	ScanNet	DL3DV	ETH3D
DL3DV-50	CroCo-B	0.126	1.430	0.399
	Gekko-B	**0.113**	**1.359**	**0.373**
DL3DV-all	CroCo-B	0.122	1.409	0.398
	Gekko-B	**0.115**	**1.366**	**0.373**

Table 5: **Co-visibility classification from the predicted channel $\hat{\mathtt{C}}$.** One unmasked forward pass, scored against depth-and-pose ground truth (↑, *b.acc.* = balanced accuracy at a single fitted threshold). Both models Base, 200k steps on the mix of Section H. Protocol in Section I.

Score	ScanNet-1500 (chance AP 0.502)			7-Scenes (chance AP 0.726)		
	AP	ROC-AUC	b.acc.	AP	ROC-AUC	b.acc.
ℓ_{CroCo} (CroCo-B$^{\text{mix}}$)	0.576	0.577	0.539	0.799	0.603	0.554
$\hat{\mathtt{C}}$ (Gekko-B$^{\text{mix}}$, ours)	**0.763**	**0.737**	**0.691**	**0.859**	**0.702**	**0.658**

Table 6: **AP by overlap tertile** on ScanNet-1500. Chance differs per tertile.

AP (↑)	low	mid	high
chance	*0.285*	*0.506*	*0.714*
ℓ_{CroCo}	0.326	0.570	0.777
$\hat{\mathtt{C}}$ (ours)	**0.609**	**0.760**	**0.849**

$\hat{\mathtt{C}}$ reaches 0.763 AP on ScanNet-1500 against 0.576 for ℓ_{CroCo} ($+0.26$ over the 0.502 chance level against $+0.07$), and it separates the classes at one global threshold with 0.691 balanced accuracy (Table 5). On 7-Scenes, where 73% of pixels are co-visible and AP compresses, ROC-AUC is clearer: 0.702 against 0.603. The margin is widest where co-visibility is hardest: on the lowest-overlap tertile $\hat{\mathtt{C}}$ gains $+0.32$ AP over chance against $+0.04$ (Table 6). It even beats the pseudo-label it was trained on: recomputing $\mathtt{C}(\mathbf{p})$ from the same model under its 90% mask gives only 0.555 AP, since a noisy per-mask estimate is a worse signal than the quantity regressed from it.

5.7 Comparison with released cross-view backbones

The comparisons above hold architecture, data and schedule fixed, which isolates the training signal but says nothing about publicly available models. We therefore probe four frozen Large backbones under one protocol, changing only the backbone (Table 7). The self-supervised rows differ in corpus and budget, so this is *not* a controlled comparison of objectives (Table 3 remains the matched evidence), and VGGT is a fully supervised $\sim$1B reference whose corpus contains ScanNet and DL3DV. Gekko-L$^{\text{mix}}$ leads both self-supervised checkpoints on every cross-view benchmark: $+13.5$ points of pose accuracy at $10°$/0.25m over CroCo v2-L, 29% lower Chamfer error in-domain and 20% on ETH3D, and less than half the frozen flow error. The gains remain specific to cross-view geometry (Section L).

5.8 Ablation studies

Ablation studies are conducted by pre-training several Base models on ScanNet to analyze the key design choices in Gekko. The different losses are defined in Section G. All models are evaluated on ScanNet-1500 for relative metric pose estimation. Results are summarized in Table 8.

Adding MAE without the relative improvement loss (row 2) barely moves the result, so the gain comes from predicting the relative improvement, not from multi-task training. A DUSt3R-style confidence loss [57] (row 3) does well on high-overlap pairs and collapses on low-overlap ones. The relative formulation beats the absolute one, patch normalization is critical on ScanNet-all, and down-weighting low-MAE pixels (Eq. 8) is worth 18.0 points while yielding sharper maps (Figure 3). Per-row analysis in Section F.

Table 7: **Frozen Large backbones under a single probing protocol.** Identical head, fine-tuning data and schedule per column. Only the frozen backbone changes. Pose: ScanNet-1500, % of pairs (↑) within 10° and the given translation threshold. Pointmap: Chamfer Overall (↓). Flow: MPI-Sintel *clean* AEPE (↓), frozen, so not comparable to published full fine-tuning figures (Section M). The self-supervised rows are *not* data-matched: they use the released checkpoints, and Table 3 is the matched comparison. *VGGT is a fully supervised reference, not a baseline.* Details in Section J. Best self-supervised in **bold**.

Frozen backbone	Sup.	ScanNet-1500 pose (↑)			Pointmap Overall (↓)			Sintel (↓)
		0.25m	0.5m	1m	ScanNet	ETH3D	DL3DV	AEPE clean
Gekko-L$^{\text{mix}}$ (ours)	self	**33.2**	**53.3**	**60.1**	**0.077**	**0.317**	**1.122**	**6.17**
CroCo v2-L [61]	self	19.7	36.2	45.3	0.109	0.394	1.359	13.08
MuM-L [37]	self	19.9	32.9	39.4	0.106	0.377	1.321	22.44
VGGT [55]	*full*	*92.3*	*98.3*	*99.0*	*0.030*	*0.240*	*1.065*	–

Table 8: **Ablation studies on ScanNet-1500.** Pose estimation accuracy (↑) at three thresholds. All models are Base, pre-trained on ScanNet. Best results in **bold**. Loss definitions in Section G.

Configuration	ScanNet-50			ScanNet-all		
	10°/.25m	10°/.5m	10°/1m	10°/.25m	10°/.5m	10°/1m
ℓ_{CroCo} (CroCo-B)	5.5	12.9	18.5	6.6	15.7	23.0
$\ell_{\text{CroCo}} + \ell_{\text{MAE}}$	6.7	15.3	20.9	6.2	14.5	21.2
ℓ_{conf}	26.7	38.7	45.8	2.2	6.1	11.2
$\ell_{\text{CroCo}} + \ell_{\text{MAE}} + \ell_{\text{AI}}$	14.3	23.0	25.2	23.3	33.6	39.4
$\ell_{\text{CroCo}} + \ell_{\text{MAE}} + \ell_{\text{AI}}$, no norm.	20.9	38.1	42.3	14.2	21.9	25.1
$\ell_{\text{CroCo}} + \ell_{\text{MAE}} + \ell_{\text{RI ratio}}$	27.2	39.0	42.7	25.7	39.6	44.4
$\ell_{\text{CroCo}} + \ell_{\text{MAE}} + \ell_{\text{RI}}$ (Gekko-B)	**29.1**	**43.3**	**47.5**	**43.7**	**54.4**	**57.6**

5.9 Generality across architectures

The gain could be an artefact of CroCo's encoder–decoder. We therefore pre-trained four more Base models from scratch on ScanNet-50 (100k steps, global batch 768), replacing one half of the architecture at a time (Table 9): a single-stream *joint self-attention* decoder, which concatenates both views' tokens with a learned per-view embedding and has no cross-attention anywhere, and a *frozen* DINOv3 [47] encoder, of which only the decoder is trained. The advantage survives both substitutions ($+17.2$ and $+18.9$ points at 10°/0.25m, against $+23.6$ with CroCo's architecture), while the CroCo baseline barely moves (5.0 to 5.7): the training signal, not the architecture, is what varies. A frozen DINOv3 with no cross-view pre-training reaches only $6.9/14.5/21.2$ on pose and 0.142 on pointmap, so the third row does not inherit its accuracy from DINOv3.

5.10 Limitations

Training cost. Gekko is more expensive to pre-train than CroCo. The extra full-resolution pass that predicts $\hat{\mathtt{C}}$ roughly doubles activation memory and so halves the per-device batch (48 vs. 96 images for Base models), which means twice the GPUs for the same global batch of 768. A matched pair of 100k-step Base pre-trainings costs $\sim 2.4\times$ more GPU-hours for Gekko. Our reference run takes ~ 25 h on 16 H100 GPUs, i.e. ~ 400 H100-hours. The cost is confined to pre-training: the extra channel is one output dimension and the MAE branch is discarded afterwards, so fine-tuning and inference cost what CroCo costs.

Where the ratio is uninformative. $\mathtt{C}(\mathbf{p})$ is a ratio, and it is unreliable wherever its denominator is small. A low $\ell_{\text{MAE}}(\mathbf{p})$ means the pixel was already predictable from the target alone, in uniform regions but equally in repeated or self-similar structure, so the reference cannot improve on it and $\mathtt{C}(\mathbf{p})$ stays near zero regardless of true co-visibility. Eq. 8 makes ℓ_{MAE} its own detector of this failure by down-weighting exactly those pixels, which Table 8 shows is worth 18.0 points. It is a soft down-weighting, not an exclusion. Explicitly masking such regions could help further.

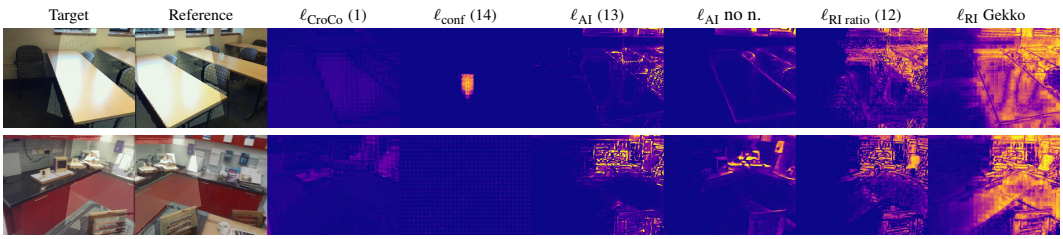


Figure 3: **Qualitative comparison of relative improvement maps $\hat{\mathrm{C}}$ across ablation configurations.** The relative improvement loss ℓ_{RI} (Eq. 8) produces sharper maps that better delineate co-visible regions, while other configurations yield unreliable predictions. The first row is ScanNet, the second 7-Scenes, which no model here saw during pre-training. How the maps are produced, and more examples in Section Q.

Table 9: **The advantage survives replacing either half of the architecture.** ScanNet-1500 pose (↑, 10° and 0.25/0.5/1m) and ScanNet pointmap Chamfer Overall (↓). All models Base, pre-trained on ScanNet-50 for 100k steps at global batch 768, with matched head budgets.

Encoder	Decoder	Pose (↑)		Pointmap (↓)	
		CroCo	Gekko	CroCo	Gekko
ViT (from scratch)	cross-attention (Table 3)	5.5 / 12.9 / 18.5	**29.1 / 43.3 / 47.5**	0.125	**0.100**
ViT (from scratch)	joint self-attention	5.7 / 9.9 / 13.9	**22.9 / 34.2 / 38.3**	0.128	0.105
DINOv3 (frozen)	cross-attention	5.0 / 10.3 / 14.7	**23.9 / 36.8 / 41.3**	0.125	**0.096**

Cross-view, frozen. Two scope limits are worth stating plainly. First, the relative improvement is a *binocular* signal and improves the tasks that need one: on single-image probes (NYUv2 depth, ADE20K segmentation, ImageNet classification), Gekko and CroCo are within about a point of each other (Section L), so we claim no general-purpose representation improvement. Second, every table here probes a *frozen* backbone, which measures what pre-training makes directly accessible rather than its value as an initialisation. Under full fine-tuning the advantage narrows unevenly: on 7-Scenes the position advantage survives and the rotation advantage largely does not, and on MPI-Sintel the matched pair ends in a tie with the released CroCo v2, pre-trained on 7.3M curated, overlap-selected image pairs, ahead of both (Section M). Our claim is accordingly the narrower one: Gekko gives better frozen cross-view features, not necessarily a better initialisation for full fine-tuning.

Data composition, dynamic scenes, and architecture. More data is not automatically better: swapping an indoor-only mix for the full 12-source mix *costs* Gekko five points on indoor ScanNet-1500 while leaving CroCo unchanged (Section K). We evaluate only static scenes: nothing in the objective assumes rigidity, but we have not tested it. Finally, Gekko inherits CroCo's restriction to image pairs and a fixed patch size, and a multi-view extension [31, 37] would have to define the relative improvement against a set of references instead of a single one, which we see as an interesting and non-trivial research direction.

6 Conclusion

Cross-view completion learns strong features for 3D vision in co-visible regions, but where the reference adds little the training signal is implicitly monocular. We showed that the relative improvement of the cross-view reconstruction error over an MAE error is a self-supervised proxy for co-visibility needing no ground-truth depth or poses, and that a network trained to predict it recovers co-visibility better than the pseudo-label it was trained on. Building on this, Gekko jointly performs cross-view completion, masked autoencoding and relative improvement prediction, supplying a binocular signal for *all* masked regions at the cost of one extra output channel. At matched architecture and data it consistently outperforms CroCo on the cross-view tasks we evaluate, and it trains directly from raw video with a simple stride curriculum, removing the 3D preprocessing prior work requires [60, 61, 36].

Acknowledgments and Disclosure of Funding

This work was supported by the Bosch Research Foundation (Bosch Forschungsstiftung) and by the European Union (ERC Advanced Grant Explorer, Funding ID #101097259). It was granted access to the HPC resources of IDRIS under the allocation 2026-AD010617525R1 made by GENCI.

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7822–7831, 2021.

[2] Honggyu An, Jin Hyeon Kim, Seonghoon Park, Jaewoo Jung, Jisang Han, Sunghwan Hong, and Seungryong Kim. Cross-view completion models are zero-shot correspondence estimators. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1103–1115, 2025.

[3] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

[4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=p-BhZSz59o4`.

[5] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *NeurIPS Datasets and Benchmarks Track*, 2021.

[6] Guillaume Bono, Leonid Antsfeld, Boris Chidlovskii, Philippe Weinzaepfel, and Christian Wolf. End-to-end (instance)-image goal navigation through correspondence as an emergent phenomenon. In *International Conference on Learning Representations*, 2024.

[7] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625, 2012.

[8] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv preprint arXiv:2001.10773*, 2020.

[9] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1050–1060, 2025.

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[11] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David Mckinnon, Yanghai Tsin, and Long Quan. Aspanformer: Detector-free image matching with adaptive span transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[13] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS dataset. In *European Conference on Computer Vision*, pages 720–736, 2018.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[16] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16739–16752, 2025.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[18] Johan Edstedt, Ioannis Athanasiadis, Mårten Wadenbäck, and Michael Felsberg. Dkm: Dense kernelized feature matching for geometry estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17765–17775, 2023.

[19] Johan Edstedt, Georg Bökman, and Zhenjun Zhao. Dedode v2: Analyzing and improving the dedode keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4245–4253, 2024.

[20] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024.

[21] Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. In *European Conference on Computer Vision*, 2026.

[22] Sven Elflein, Qunjie Zhou, Sérgio Agostinho, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16774–16784, 2025.

[23] Miao Fan, Mingrui Chen, Chen Hu, and Shuchang Zhou. Occ^{2}net: Robust image matching based on 3d occupancy estimation for occluded regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9652–9662, 2023.

[24] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Visual correspondence hallucination. In *International Conference on Learning Representations*, 2022.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[27] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021.

[28] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.

[29] Shinjeong Kim, Marc Pollefeys, and Daniel Barath. Learning to make keypoints sub-pixel accurate. In *European Conference on Computer Vision*, pages 413–431. Springer, 2024.

[30] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024.

[31] Wenyu Li, Sidun Liu, Peng Qiao, Yong Dou, and Tongrui Hu. Muskie: Multi-view masked image modeling for 3d vision pre-training. *arXiv preprint arXiv:2511.18115*, 2025.

[32] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2022.

[33] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In *International Conference on Learning Representations*, 2026.

[34] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17627–17638, 2023.

[35] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024.

[36] Thibaut Loiseau, Guillaume Bourmaud, and Vincent Lepetit. Alligat0r: Pre-training through co-visibility segmentation for relative camera pose regression. In *Advances in Neural Information Processing Systems*, 2025.

[37] David Nordström, Johan Edstedt, Fredrik Kahl, and Georg Bökman. Mum: Multi-view masked image modeling for 3d vision. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21736–21747, 2026.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[39] Guilherme Potje, Felipe Cadar, André Araujo, Renato Martins, and Erickson R Nascimento. Xfeat: Accelerated features for lightweight image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2682–2691, 2024.

[40] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[41] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021.

[42] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *IEEE/CVF International Conference on Computer Vision*, pages 10912–10922, 2021.

[43] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[44] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017.

[45] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, pages 746–760, 2012.

[47] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, et al. DINOv3. *Transactions on Machine Learning Research*, 2026.

[48] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024.

[49] Zikai Song, Run Luo, Junqing Yu, Yi-Ping Phoebe Chen, and Wei Yang. Compact transformer tracker with correlative masked modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2321–2329, 2023.

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[51] Dongli Tan, Jiang-Jiang Liu, Xingyu Chen, Chao Chen, Ruixin Zhang, Yunhang Shen, Shouhong Ding, and Rongrong Ji. Eco-tr: Efficient correspondences finding via coarse-to-fine refinement. In *European Conference on Computer Vision*, pages 317–334. Springer, 2022.

[52] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *International Conference on Learning Representations*, 2022.

[53] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10247–10266, 2023.

[54] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision*, pages 78–89, 2025.

[55] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[56] Qing Wang, Jiaming Zhang, Kailun Yang, Kunyu Peng, and Rainer Stiefelhagen. Matchformer: Interleaving attention in transformers for feature matching. In *Proceedings of the Asian Conference on Computer Vision*, pages 2746–2762, 2022.

[57] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024.

[58] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. TartanAir: A dataset to push the limits of visual SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4909–4916, 2020.

[59] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. Efficient loftr: Semi-dense local feature matching with sparse-like speed. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21666–21675, 2024.

[60] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

[61] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023.

[62] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[63] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Information Processing Systems*, volume 37, pages 21875–21911, 2024.

[64] Lihe Yang, Shang-Wen Li, Yang Li, Xinjie Lei, Dong Wang, Abdelrahman Mohamed, Saining Xie, Hengshuang Zhao, Kaiming He, and Hu Xu. In pursuit of pixel supervision for visual pre-training. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 31974–31984, 2026.

[65] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[66] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[67] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *International Conference on Learning Representations*, 2025.

[68] Haojie Zhao, Dong Wang, and Huchuan Lu. Representation learning for visual object tracking by masked appearance transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18696–18705, 2023.

[69] Xiaoming Zhao, Xingming Wu, Weihai Chen, Peter CY Chen, Qingsong Xu, and Zhengguo Li. Aliked: A lighter keypoint and descriptor extraction network via deformable transformation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–16, 2023.

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017.

[71] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=ydopy-e6Dg`.

[72] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics*, 37 (4), 2018.

[73] Shengjie Zhu and Xiaoming Liu. Pmatch: Paired masked image modeling for dense geometric matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21909–21918, 2023.

[74] Lojze Zust, Yohann Cabon, Juliette Marrie, Leonid Antsfeld, Boris Chidlovskii, Jerome Revaud, and Gabriela Csurka. Panst3r: Multi-view consistent panoptic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5856–5886, 2025.

A Broader impact statement

This work contributes a self-supervised pre-training method for 3D vision. The primary applications are in 3D reconstruction, visual localization, and camera pose estimation, which benefit fields such as robotics, autonomous navigation, and augmented reality. As with any method that improves visual understanding, potential dual-use risks include surveillance applications. However, the method operates on pre-training and does not introduce capabilities beyond those already available in existing systems.

B CroCo and MAE formalism

CroCo. Given a target image $\mathtt{I}_\mathrm{T}$ and a reference image $\mathtt{I}_\mathrm{R}$ of the same scene from different viewpoints, a Cross-view Completion network F_{CroCo} reconstructs the target image $\mathtt{I}_\mathrm{T}$ from a masked version of itself and the reference image $\mathtt{I}_\mathrm{R}$:

$$\hat{\mathtt{I}}_{\mathrm{T}|\mathrm{R}} = F_{\text{CroCo}}\left(\mathtt{M} \odot \mathtt{I}_\mathrm{T}, \mathtt{I}_\mathrm{R}\right), \tag{9}$$

where $\mathtt{M}$ masks out a random subset (90% in practice) of the target input patches. A ViT encoder first encodes the reference patches and the unmasked target patches separately. A cross-attention based decoder then processes reference and target tokens to warp information from the reference into the target representation. A lightweight pixel decoder reconstructs the target image $\hat{\mathtt{I}}_{\mathrm{T}|\mathrm{R}}$. CroCo is trained to minimize the total reconstruction error on masked target pixels: $\mathcal{L}_{\text{CroCo}} = \sum_{\mathbf{p}\in\Omega_\mathtt{M}} \ell_{\text{CroCo}}\left(\mathbf{p}\right)$, where $\Omega_\mathtt{M}$ is the set of masked target pixel locations and ℓ_{CroCo} is defined in Eq. 1.

MAE. A Masked Autoencoder F_{MAE} reconstructs $\mathtt{I}_\mathrm{T}$ from a masked version of itself *only*:

$$\hat{\mathtt{I}}_\mathrm{T} = F_{\text{MAE}}\left(\mathtt{M} \odot \mathtt{I}_\mathrm{T}\right), \tag{10}$$

where $\mathtt{M}$ masks out a random subset (75% in monocular MAE, we perform 90% to get the same mask as CroCo and be able to compute the loss and the relative improvement) of the target input patches. MAE is trained to minimize $\mathcal{L}_{\text{MAE}} = \sum_{\mathbf{p}\in\Omega_\mathtt{M}} \ell_{\text{MAE}}\left(\mathbf{p}\right)$, where ℓ_{MAE} is defined in Eq. 2.

C Training details

Stride curriculum. The two curriculum variants of Section 5.4 share N=5k steps between increments and differ only in the increment itself: Δk=2 for the *-50* variant and Δk=10 for the *-all* variant. All other hyperparameters are identical to the preprocessed-data runs. The schedule is fixed so that the comparison against preprocessed baselines is controlled. An adaptive per-sequence rule (for instance increasing the stride on plateau) would make the approach fully automatic.

Pre-training. All models are trained from scratch with a batch size of 768 and a learning rate of 1.5×10^{-4} for 100k steps. A cosine learning rate decay with a linear warmup of 5000 steps is used. Following CroCo [60], masking ratio is 90% and patches are normalized to zero mean and unit variance. Images are resized to a maximum dimension of 512 pixels while maintaining aspect ratio.

Fine-tuning for relative metric pose estimation. A simple MLP head is added on top of the frozen pre-trained decoder features, trained with a homoscedastic loss [28] for 20k steps with a batch size of 768 and a constant learning rate of 1×10^{-4}. Fine-tuning uses the *all* overlap variant to leverage the full range of viewpoint changes.

Fine-tuning for pointmap regression. A DPT head [40] is fine-tuned on top of frozen pre-trained features using the confidence-aware regression loss from DUSt3R [57]. Fine-tuning uses the ScanNet-all dataset with the same batch size and learning rate as pose estimation.

D Architecture details

Following CroCo [60], all models use a Vision Transformer (ViT) architecture [17] in two sizes:

- **Base**: 12-layer ViT encoder and 8-layer transformer decoder with embedding sizes of 768 and 512, pre-trained on both variants of ScanNet.
- **Large**: 24-layer ViT encoder and 12-layer transformer decoder with embedding sizes of 1024 and 768, pre-trained on ScanNet-all (Table 3) and, for Gekko-L$^{\text{mix}}$, on the mix of Section H.

E CroCo qualitative reconstructions



Figure 4: **Qualitative CroCo reconstructions.** Reconstructions are sharper in co-visible regions where the network warps information from the reference image. Conversely, reconstructions are blurred in non-co-visible regions where the reference provides little information (best viewed zoomed-in).

F Detailed ablation analysis

This section provides a detailed discussion of each ablation row in Table 8.

Impact of the relative improvement loss. Adding the MAE reconstruction branch without the relative improvement loss ℓ_{RI} (row 2) provides only marginal improvement over the CroCo baseline (20.9% vs. 18.5% at $10°$/1m). The key contribution of Gekko lies in the relative improvement prediction, not merely in jointly training with MAE.

Confidence prediction vs. relative improvement. The confidence-aware loss performs well on high-overlap pairs (45.8% at $10°$/1m on ScanNet-50) but fails on low-overlap pairs (11.2% on ScanNet-all). As illustrated in Figure 3, the confidence maps lack the sharp co-visibility boundaries that ℓ_{RI} achieves.

Relative vs. absolute improvement. Regressing the absolute improvement $\ell_{\text{MAE}} - \ell_{\text{CroCo}}$ underperforms the relative formulation, particularly on ScanNet-all (39.4% vs. 57.6%). The relative improvement produces sharper maps that better capture co-visibility boundaries (Figure 3).

Impact of patch normalization. Without normalizing target patch tokens, performance drops significantly on ScanNet-all (25.1% vs. 39.4%). Removing normalization leads to noisier predictions in uniform regions where MAE predicts a near-constant value, yielding unreliable relative improvement estimates.

Loss formulation details. Directly regressing the ratio underperforms the final formulation (Eq. 8) that multiplies the prediction by the denominator, improving from 25.7% to 43.7% at $10°$/.25m on ScanNet-all.

G Ablation study: technical details

The different losses considered in the ablation study (Table 8) are defined as follows:

1. $$\ell_{\text{RI}}(\mathbf{p}) = \left(sg\left[\ell_{\text{MAE}}(\mathbf{p}) - \ell_{\text{CroCo}}(\mathbf{p})\right] - sg\left[\ell_{\text{MAE}}(\mathbf{p})\right]\hat{\mathtt{C}}(\mathbf{p})\right)^2 \tag{11}$$
   is the relative-improvement loss of Gekko (see Eq. 3).

2. $$\ell_{\text{RI ratio}}(\mathbf{p}) = \left(sg\left[\frac{\ell_{\text{MAE}}(\mathbf{p}) - \ell_{\text{CroCo}}(\mathbf{p})}{\ell_{\text{MAE}}(\mathbf{p})}\right] - \hat{\mathtt{C}}(\mathbf{p})\right)^2 \tag{12}$$
   is a relative-improvement loss that reweights the previous loss as follows: $\left(\frac{1}{sg[\ell_{\text{MAE}}(\mathbf{p})]}\right)^2 \ell_{\text{RI}}(\mathbf{p}) = \ell_{\text{RI ratio}}(\mathbf{p})$. This reweighting up-weights pixels where the MAE loss is low, which often correspond to uniform regions that yield noisy pseudo-labels. In practice, ℓ_{RI} significantly outperforms $\ell_{\text{RI ratio}}$.

3. $$\ell_{\text{AI}}(\mathbf{p}) = \left(sg\left[\ell_{\text{MAE}}(\mathbf{p}) - \ell_{\text{CroCo}}(\mathbf{p})\right] - \hat{\mathtt{C}}(\mathbf{p})\right)^2 \tag{13}$$
   is an absolute-improvement loss where $\hat{\mathtt{C}}(\mathbf{p})$ learns to predict the difference between ℓ_{CroCo} and ℓ_{MAE}.

4. $$\ell_{\text{conf}}(\mathbf{p}) = \left(1 + \exp\left(\hat{\mathtt{C}}(\mathbf{p})\right)\right)\left\|\mathtt{I}_{\mathtt{T}}(\mathbf{p}) - \hat{\mathtt{I}}_{\mathtt{T|R}}(\mathbf{p})\right\|_2^2 - 0.2\ln\left(1 + \exp\left(\hat{\mathtt{C}}(\mathbf{p})\right)\right) \tag{14}$$
   is a confidence-aware loss where $\hat{\mathtt{C}}(\mathbf{p})$ learns to predict the precision (*i.e.* inverse variance) that is expected to be high in co-visible regions and low in non-co-visible regions.

H The 12-source raw-video mix

Two experiments (Section 5.6 and Section 5.7) use a mix of twelve video sources instead of the curated Cub3 pairs: ScanNet [13], ScanNet++ [65], ARKitScenes [5], RealEstate10K [72] and EPIC-KITCHENS [14] (real indoor); Hypersim [42] (synthetic indoor); CO3Dv2 [41] and Objectron [1] (object-centric); DL3DV [35] and KITTI-360 [32] (real outdoor); and VirtualKITTI2 [8] and TartanAir [58] (synthetic outdoor).

Every source is consumed as *raw video* through the stride curriculum of Section 5.4: pairs are formed by sampling two frames of the same clip at a stride drawn from the current curriculum window. No structure-from-motion, depth, camera pose or overlap estimate is used at any point, so adding a source costs only the frames themselves. Sources are sampled with fixed weights, and validation uses a scene-disjoint held-out split of each source. Unless stated otherwise, mix models are trained with the same optimiser, resolution and global batch size as the ScanNet models of Section C.

I Co-visibility probing protocol

For Table 5 we run each model once per pair with *no masking* and read the fourth output channel $\hat{\mathtt{C}}$ directly. There is no post-processing, no test-time augmentation and no fitting of the network. Ground truth is computed on the fly: a target pixel is labelled co-visible if its back-projected 3D point reprojects inside the reference image and the two depths agree within 5%. Pixels whose reprojection falls on a hole in the reference depth map are *excluded* rather than counted as negatives, which would otherwise reward a model for predicting low co-visibility on missing data.

We evaluate on all 1500 pairs of ScanNet-1500 and on 558 retrieval pairs of 7-Scenes [45]. Neither benchmark contributes a scene to the mix of Section H. Average precision and ROC-AUC are threshold-free. For balanced accuracy we fit a single one-dimensional logistic regression per dataset and per score over all evaluated pixels and threshold it at $p = 0.5$. The question is whether *one* global threshold separates the classes at all, so the threshold is fitted on the same pixels it is reported on. The CroCo baseline has no fourth channel, so it is scored by $-\ell_{\text{CroCo}}(\mathbf{p})$ from a masked forward pass.

The tertile analysis of Table 6 sorts the 1500 pairs by their true co-visible fraction and splits them into three equal groups. Chance average precision is the positive rate within each group and therefore differs per tertile.

Two remarks. First, the pseudo-label figure quoted in Section 5.6 (0.555 AP) is $\mathtt{C}(\mathbf{p})$ recomputed from the *same trained model* under its 90% training mask, averaged over mask draws. It is not directly comparable to the 0.74 of Figure 2, which is measured on a different set of 100 ScanNet-50 pairs using separately pre-trained CroCo and MAE models (Section O). Second, restricting the evaluation to SIFT keypoints raises every score, and raises the pseudo-label far more than the prediction ($\mathtt{C}(\mathbf{p})$ $0.555 \to 0.634$, against $\hat{\mathtt{C}}$ $0.763 \to 0.790$ and ℓ_{CroCo} $0.576 \to 0.587$): the textureless-region limitation of Section 5.10 showing up in the raw ratio rather than in the learned channel.

J Probing protocol for Table 7

All rows of Table 7 share the head architecture, the fine-tuning data and the step budget of the corresponding main-paper experiment. Only the frozen backbone is swapped. Gekko-L$^{\text{mix}}$ is pre-trained for 500k steps on the mix of Section H. Parameter counts: $\sim$420M for Gekko-L$^{\text{mix}}$ and CroCo v2-L, which share the ViT-L encoder and Base decoder, $\sim$390M for MuM-L, and $\sim$1B for VGGT. Pose uses the MLP head of Section C on frozen decoder features, pointmap uses the DPT head, optical flow uses a DPT flow head with the tiled inference of CroCo-Flow at 0.9 overlap on the MPI-Sintel [7] `subval` split.

Two caveats. The self-supervised rows are not data-matched: CroCo v2 is pre-trained on 7.3M curated image pairs selected for visual overlap (2M Habitat and 5.3M real) and MuM on a corpus that includes ImageNet-1K, whereas Gekko-L$^{\text{mix}}$ sees only the raw video of Section H. And MuM's frozen rows use a learning rate of 10^{-4}, since the rate used for the other rows does not train its head at all. VGGT is reported for scale only: it is trained with explicit camera and depth supervision, and both ScanNet and DL3DV are in its training corpus, so three of its four columns are in-domain for it.

K Data scaling on the raw-video mix

To test whether the relative improvement signal keeps paying off as data grows, we pre-train matched pairs of Base models on nested, seeded, scene-level subsets of the mix of Section H, from 1% to 100% of its scenes. Sources, sampling weights, resolution, global batch size (768) and step count (100k) are identical across all rows, as is the fine-tuning budget of every head.

Gekko leads at every scale, by between $+7.4$ and $+24.1$ points at the strictest pose threshold. Two honest caveats. Above 50% both methods dip on pose at a fixed 100k-step budget, which indicates the backbones are undertrained rather than that data hurts. Extending the best mix model to 200k steps recovers $39.4/56.1/61.1$ and a pointmap Overall of 0.092. And composition matters more than size on an indoor benchmark: an indoor-only mix gives Gekko $39.4/59.2/65.5$ against $34.4/51.5/56.5$ for the full mix, while CroCo is essentially flat ($13.7/29.9/38.6$ against $15.1/30.5/39.2$). Adding outdoor and object-centric video buys generality at a measurable in-domain cost.

Table 10: **Data scaling.** ScanNet-1500 pose (↑, at 10°/0.25 0.5 1m) and ScanNet pointmap Chamfer Overall (↓) as a function of the fraction of mix scenes used for pre-training. Best per column in **bold**.

Pre-training data	Pose (↑)		Pointmap (↓)	
	CroCo-B	Gekko-B	CroCo-B	Gekko-B
1%	7.3 / 15.7 / 20.7	**15.1 / 24.7 / 29.0**	0.144	**0.140**
5%	17.7 / 32.9 / 39.3	**25.1 / 40.8 / 46.3**	0.126	**0.121**
10%	15.2 / 26.6 / 33.4	**37.0 / 54.1 / 58.7**	0.118	**0.110**
25%	17.4 / 33.7 / 42.0	**41.5 / 57.5 / 63.2**	0.108	**0.103**
50%	22.5 / 39.3 / 46.9	**37.9 / 56.5 / 61.4**	0.104	**0.101**
100%	15.1 / 30.5 / 39.2	**34.4 / 51.5 / 56.5**	0.102	**0.096**

L Single-image probes

The relative improvement is a binocular signal, so we do not expect it to help tasks that see one image. It does not, and it does not hurt either. Under the frozen protocol of the main paper with a DPT head, monocular depth on NYUv2 [46] (REL ↓ / δ_1 ↑) and semantic segmentation on ADE20K [70] (mIoU ↑ / pixel accuracy ↑) give:

Table 11: **Single-image probes are a tie.** Frozen backbone, DPT head, matched schedules. All models Base.

Pre-training	NYUv2 depth		ADE20K segmentation	
	CroCo-B	Gekko-B	CroCo-B	Gekko-B
ScanNet-50	**0.181 / 73.5**	0.191 / 72.3	17.6 / 71.2	17.6 / **71.5**
ScanNet-all	0.200 / **70.9**	0.200 / 70.5	**17.6 / 71.3**	16.8 / 70.9
12-source mix, 200k	0.147 / 80.8	**0.143 / 82.2**	**20.8** / 73.6	20.2 / **73.8**

The same holds for ImageNet-1K [15] classification with an attentive probe (a learned query cross-attending the frozen patch tokens, then a linear classifier, with 50 epochs at batch 2048): 61.3 top-1 for Gekko-B$^{\text{mix}}$ against 61.0 for CroCo-B$^{\text{mix}}$, a tie. On the eight dense Taskonomy [66] tasks (Table 12, standard 1k-image transfer protocol, masked L1 on the full test split), the matched Base pair splits the columns five to three, by small margins either way. Against the released Large backbones of Table 7, Gekko-B$^{\text{mix}}$ is best on 2D texture and occlusion edges and behind on the other six, mostly to MuM-L, which is three times larger and whose pre-training corpus includes ImageNet-1K. We conclude that the relative improvement objective buys cross-view geometry and nothing else.

Table 12: **Taskonomy, frozen backbones.** Masked L1 $\times 1000$ (↓) on the test split, identical DPT head and schedule in every row. Best per column in **bold**.

Frozen backbone	Normals	Depth	Reshading	Curvature	2D edges	Occl. edges	2D kpts	3D kpts
CroCo-B$^{\text{mix}}$	65.5	33.9	107.4	45.0	3.54	0.76	0.13	46.4
Gekko-B$^{\text{mix}}$ (ours)	67.4	34.2	107.1	44.1	**2.89**	**0.58**	0.35	46.2
CroCo v2-L (released)	59.4	31.0	95.9	47.5	3.55	0.67	**0.11**	43.6
MuM-L (released)	**57.8**	**26.9**	**88.8**	**42.4**	3.17	0.59	0.49	**43.2**

M CroCo's flow and pose benchmarks: frozen and full fine-tuning

Every table in the main paper probes a frozen backbone. CroCo also reports optical flow on MPI-Sintel [7] and relative pose on 7-Scenes [45] after fine-tuning the whole network, which measures a different property, a model's value as an initialisation, and ranks models differently. Table 13 reports both protocols on these two benchmarks.

Optical flow. Full fine-tuning follows CroCo-Flow's recipe and sample budget: 110k steps at effective batch 64, i.e. 7.0M of their 7.2M sample-passes. The matched Base pair, 17% apart in Gekko's favour when frozen, ends in a tie (better on clean, worse on final), and CroCo v2-L is clearly ahead of everything. Sintel flow is estimated between consecutive frames, i.e. on high-overlap pairs, which is the regime CroCo v2's 7.3M overlap-selected pre-training pairs are curated for. The

Table 13: **Frozen backbone versus full-network fine-tuning on CroCo's benchmarks.** MPI-Sintel `subval` AEPE clean / final (↓) and 7-Scenes median position (cm) / orientation (°) error (↓). *Frozen*: head-only protocol of the main paper. *Full FT*: whole network, at CroCo-Flow's sample budget for Sintel and at 224×224 for 6k steps for 7-Scenes. Every row of a column uses the same recipe. Better of the matched Base pair in **bold**.

Backbone	MPI-Sintel AEPE (↓)		7-Scenes pos. / rot. (↓)	
	frozen	full FT	frozen	full FT
CroCo-B$^{\text{mix}}$	8.21 / 7.93	2.21 / **2.59**	31.9 / 9.36	7.70 / 2.08
Gekko-B$^{\text{mix}}$ (ours)	**6.80 / 7.27**	**2.02** / 2.72	**23.2 / 6.74**	**6.16 / 1.90**
CroCo v2-L (released)	13.08 / 10.77	1.48 / 2.15	33.8 / 8.77	5.77 / 1.60
MuM-L (released)	22.44 / 21.69	2.59 / 3.20	28.5 / 9.73	5.29 / 1.60
CroCo v2 ViT-B/Small, published	–	*1.76 / 2.30*	–	–
CroCo v2 ViT-L/Base, published	–	*1.43 / 1.99*	–	–
CroCo v1, published	–	–	–	*5.0 / 3.46*

figure it publishes for our encoder and decoder sizes (ViT-B/Small) is $1.76/2.30$. It is also a validation of our pipeline: the released CroCo v2 checkpoint reaches 1.48 in our hands against 1.43 published, a 3% gap explained by four documented deviations (110k of 112.5k steps, effective batch 64 rather than 8 with the learning rate scaled linearly, bf16 instead of fp32, and the last checkpoint instead of the best on validation). MuM publishes no Sintel result, so its row is our own run under the same recipe.

Relative pose on 7-Scenes. Fine-tuned, Gekko-B$^{\text{mix}}$ is 20% better than CroCo-B$^{\text{mix}}$ on position and 9% on orientation, against 27% and 28% frozen. The released ViT-L models are ahead of both Base models once fine-tuned. CroCo specifies only that the whole network is fine-tuned at 224×224, so the optimiser, batch size and schedule are ours and identical across rows.

N Pointmap regression: full metrics

Table 14: **Pointmap regression, full metrics.** Accuracy, Completeness, and Overall metrics (↓) for models pre-trained on DL3DV with curriculum and fine-tuned on ScanNet-all. Gekko consistently outperforms CroCo across all benchmarks, including out-of-domain evaluation on DL3DV and ETH3D. Best result per group in **bold**.

Pre-training	Model	ScanNet			DL3DV			ETH3D		
		Acc.	Comp.	Over.	Acc.	Comp.	Over.	Acc.	Comp.	Over.
DL3DV-50	CroCo-B	0.128	0.125	0.126	1.278	1.583	1.430	0.400	0.399	0.399
	Gekko-B	**0.118**	**0.108**	**0.113**	**1.240**	**1.477**	**1.359**	**0.379**	**0.367**	**0.373**
DL3DV-all	CroCo-B	0.123	0.121	0.122	1.268	1.549	1.409	0.399	0.396	0.398
	Gekko-B	**0.119**	**0.110**	**0.115**	**1.230**	**1.503**	**1.366**	**0.376**	**0.369**	**0.373**

O Details on the computation of the histograms in Figure 2

The histograms in Figure 2 are computed on 100 image pairs from the ScanNet-50 set. For each pair, ground-truth co-visibility labels from Cub3 [36] are used: a pixel is considered co-visible if its 3D point projects within the image boundaries of the other view and the depth is consistent (within 5% tolerance). Both a pre-trained CroCo model and a pre-trained MAE model are run on each pair to obtain per-pixel reconstruction errors $\ell_{\text{CroCo}}(\mathbf{p})$ and $\ell_{\text{MAE}}(\mathbf{p})$, and the relative improvement $\mathbf{C}(\mathbf{p})$ is computed according to Eq. 3.

P PCA visualization of encoder features

To qualitatively assess the learned representations, Figures 5 and 6 visualize the first three principal components of the encoder output tokens, mapped to RGB channels, for both CroCo and Gekko. The

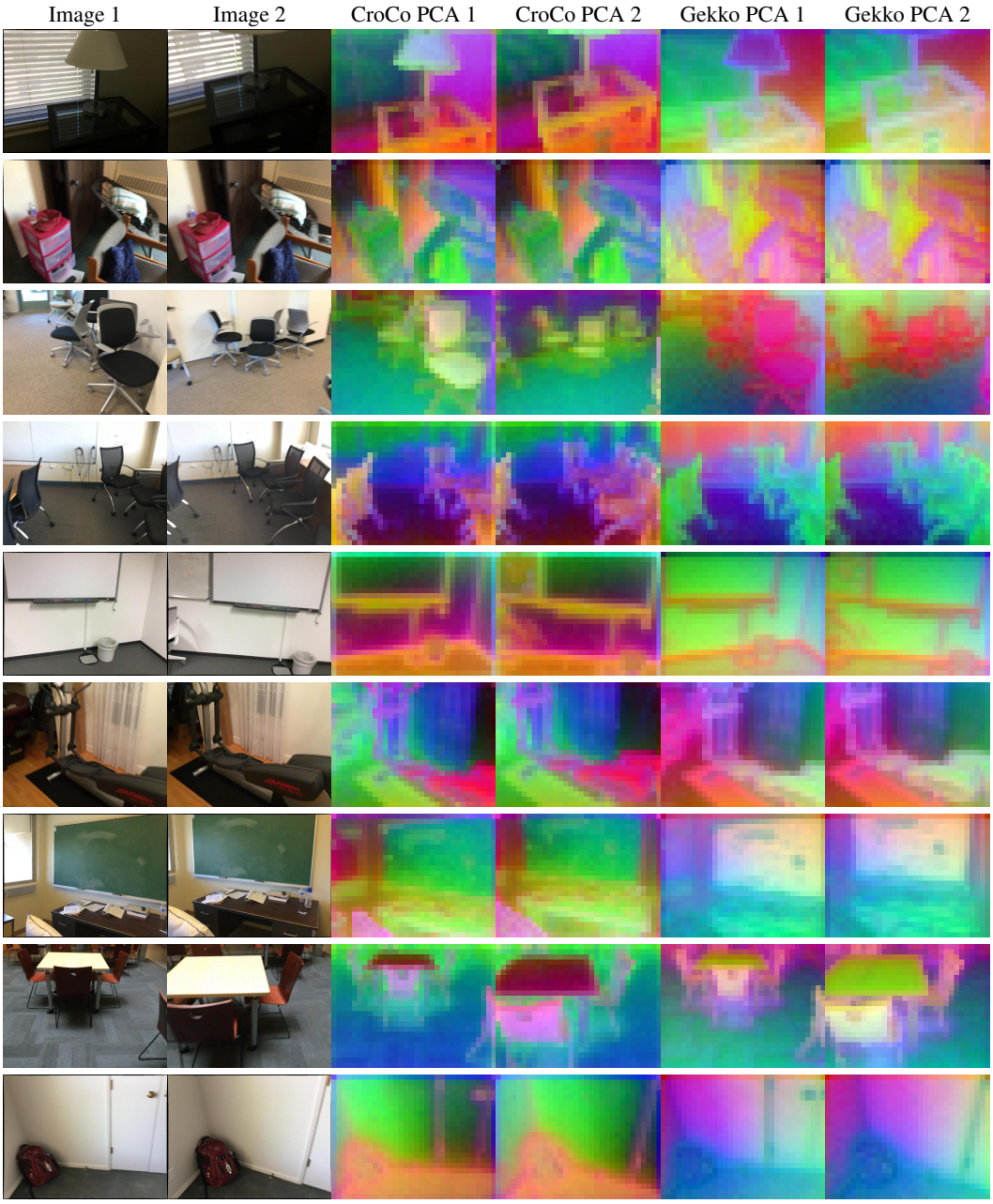


Figure 5: **PCA visualization of encoder features for CroCo and Gekko.** For each image pair, the three principal components of the encoder output tokens are mapped to RGB channels.

PCA is computed jointly over both views of each pair so that colors are directly comparable across images. Consistent colors in co-visible regions indicate that the encoder maps corresponding 3D points to similar feature vectors, a desirable property for downstream geometric tasks. Both models are the matched Base pair pre-trained on ScanNet-50. Figure 5 shows ScanNet pairs. Figure 6 adds pairs from DL3DV, Cambridge Landmarks, ETH3D and 7-Scenes, none of which either model saw during pre-training.

Q Additional qualitative results

How the maps are produced. In Figures 3 and 7, each column after the first two is a separately pre-trained Base model, using the loss named in its header. These are the models of Table 8, and nothing is shared between columns. The map is the fourth output channel of the pixel head, read from

a single forward pass with *no masking*, so the target is fully visible and attends to the full reference, with no post-processing and no ground truth involved. Two columns differ. The CroCo baseline has no fourth channel, so the ℓ_{CroCo} column shows its per-pixel reconstruction error (Eq. 1) from a masked forward pass, the quantity of Figure 2 (left), and the ℓ_{conf} column shows the predicted DUSt3R-style confidence. The green overlay on the target and reference images is the ground-truth co-visibility, shown for reference only.

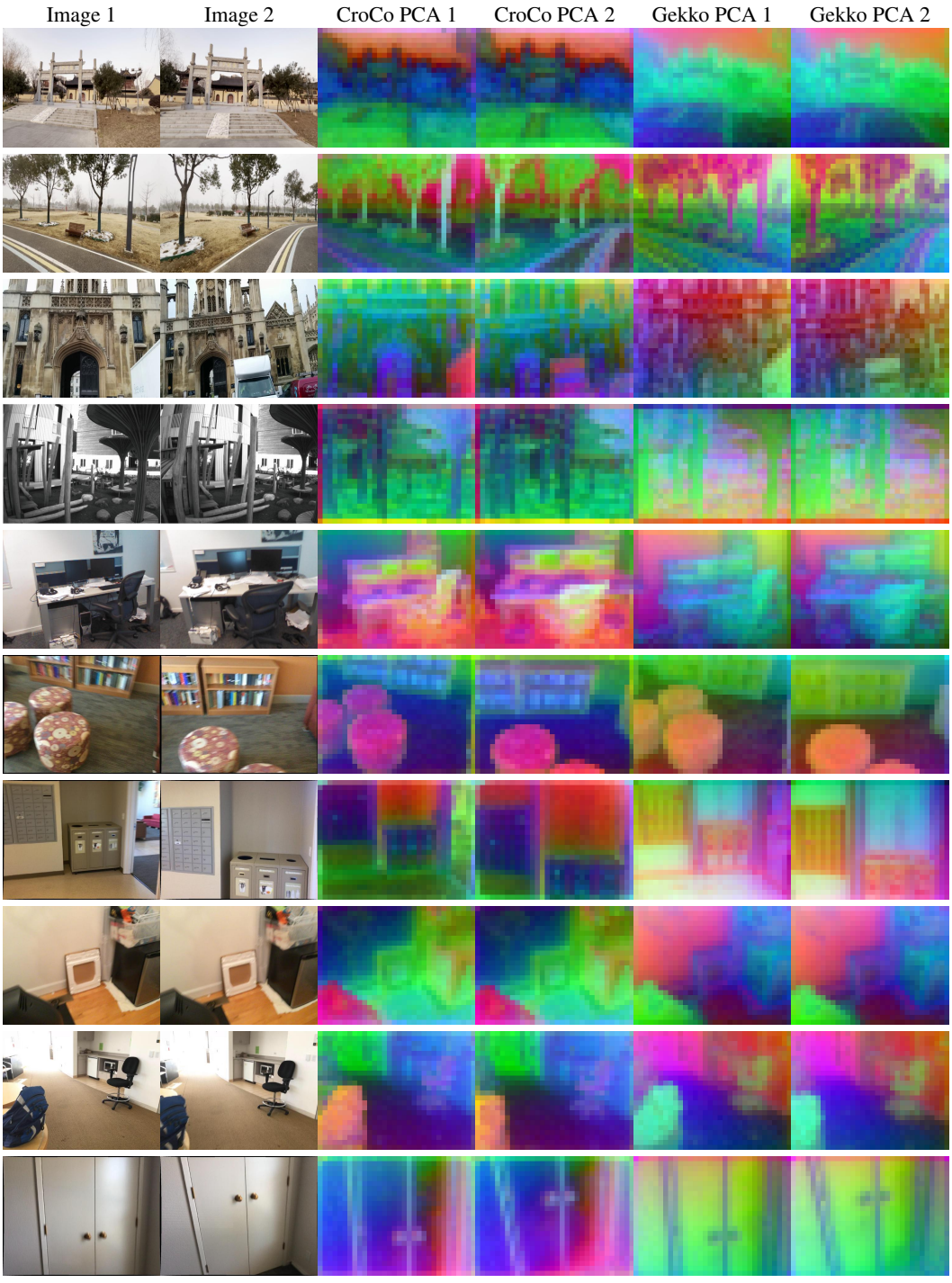


Figure 6: **PCA visualization of encoder features on other domains.** Same visualization as Figure 5. The first five rows are pairs from DL3DV (two outdoor scenes), Cambridge Landmarks, ETH3D and 7-Scenes. The remaining rows are ScanNet. Both models here are pre-trained on ScanNet alone, so the first five rows are out of domain for each of them, yet Gekko still assigns matching colors to corresponding surfaces across the viewpoint change.

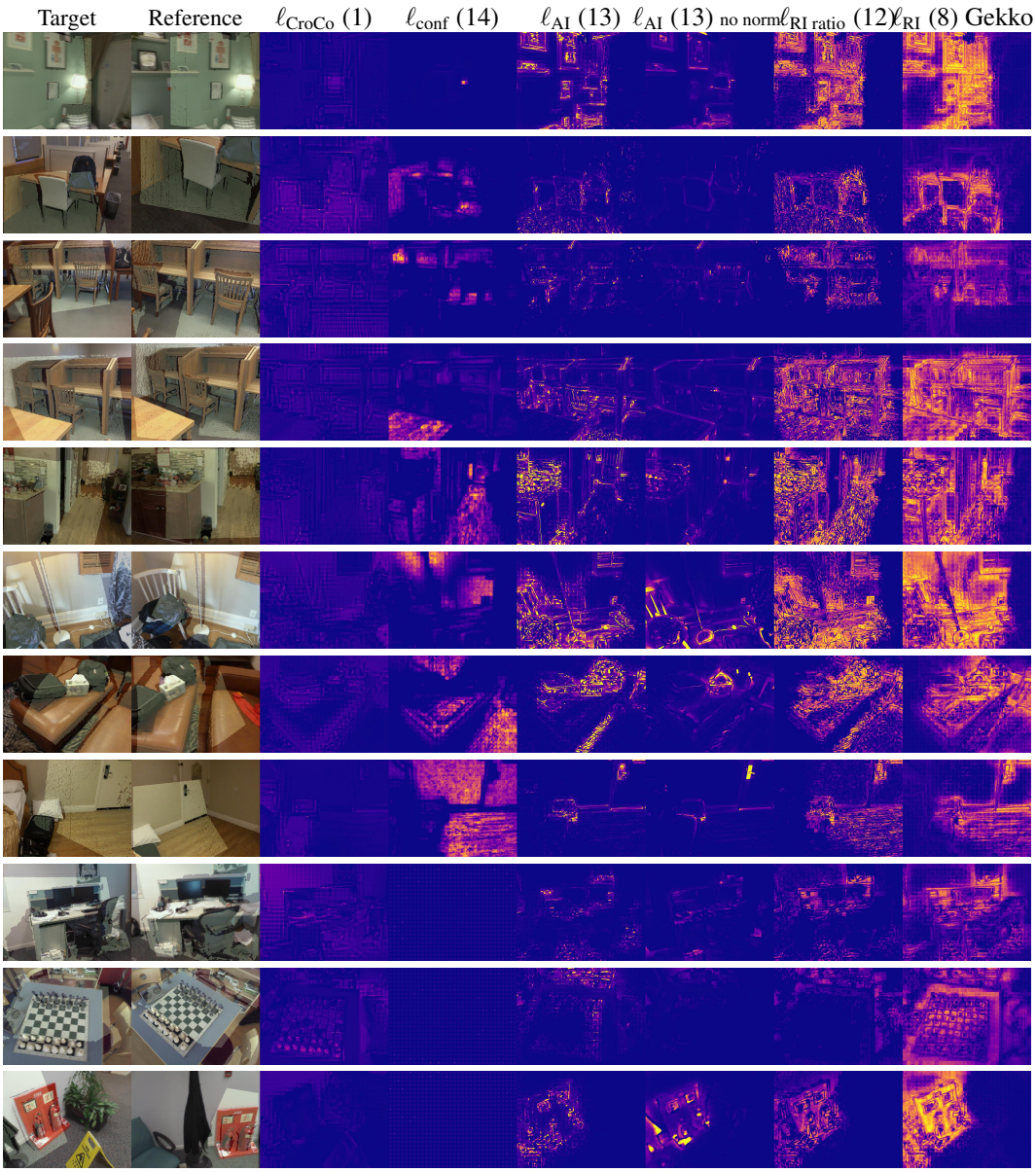


Figure 7: **Qualitative comparison of relative improvement maps $\hat{\mathbb{C}}$ across ablation configurations.** The relative improvement loss ℓ_{RI} (Eq. 8) produces sharper maps that better delineate co-visible regions, while other configurations yield unreliable predictions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction list three contributions (relative improvement as a co-visibility proxy together with the Gekko framework, consistent improvements over CroCo and released cross-view backbones, and curriculum pre-training from raw video) that are all supported by the experiments in Section 5. Section 5.10 states explicitly that the gains are specific to cross-view geometry.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5.10 discusses training cost, noisy signal in uniform regions, and architecture constraints inherited from CroCo.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results or proofs. The contribution is empirical.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 specifies architectures, training hyperparameters (batch size, learning rate, schedule, masking ratio), datasets, fine-tuning details, and evaluation protocols. All loss formulations are given in closed form.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, configuration files and pre-trained weights are publicly released (`https://github.com/thibautloiseau/gekko`, `https://huggingface.co/thibautloiseau/gekko-vitl-500k`), with instructions for data preparation, pre-training, fine-tuning and evaluation. All datasets used are publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 and Section C report batch size, learning rate, schedule, masking ratio, image resolution, number of training steps, curriculum constants and fine-tuning hyperparameters. The protocols of the additional benchmarks are given in the corresponding appendix sections.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported. Each pre-training run is computationally expensive, making multiple runs impractical. This is consistent with prior work in the same area (CroCo, DUSt3R).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5.10 reports the GPU type (16×NVIDIA H100), the per-device batch size, the wall-clock time and the total GPU-hours of a full pre-training run, together with the measured cost relative to CroCo. Per-task fine-tuning budgets are given in Section C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets and does not involve human subjects, deception, or harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A broader impact statement in Section A discusses positive applications (robotics, navigation, augmented reality) and potential dual-use risks (surveillance).

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The released models are pre-trained feature extractors for 3D vision and do not pose a high risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (ScanNet, DL3DV, ETH3D, the twelve video sources of Section H, NYUv2, ADE20K, ImageNet-1K, Taskonomy, MPI-Sintel and 7-Scenes), methods (CroCo, MAE, DUSt3R) and released models (CroCo v2, MuM, DINOv3, VGGT) are cited and used for research under their respective licenses. Our code derives from CroCo and keeps its CC BY-NC-SA 4.0 license.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released code and pre-trained weights are documented (usage, dataset preparation, configuration files) and distributed under CC BY-NC-SA 4.0. No new dataset is introduced.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core method.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.